

Научная статья

УДК 615.07:004.8

<https://doi.org/10.37493/2307-910X.2025.2.5>

Разработка методов выявления аномалий сетевого трафика с использованием алгоритмов машинного обучения

Виталий Геннадьевич Лапин^{1*}, Иван Алексеевич Гальвас², Иван Владимирович Головин³,
Егор Александрович Кеньков⁴, Максим Николаевич Васильев⁵

^{1,2,3,4} Северо-Кавказский федеральный университет (д. 1, ул. Пушкина, Ставрополь, 355017, Россия)

⁵ МИРЭА - Российский технологический университет (д. 78, пр. Вернадского, Москва, 119454, Россия)

¹ vitlx@yandex.ru; <https://orcid.org/0000-0002-0611-7002>

² norra7@yandex.ru

³ ivangolovin085@gmail.com; <https://orcid.org/0009-0008-0512-7242>

⁴ kenkov2045@yandex.ru; <https://orcid.org/0009-0005-3078-1412>

⁵ kesumen_2000@mail.ru

*Автор, ответственный за переписку

Аннотация. В условиях стремительного роста объемов сетевого трафика и усложнения структуры киберугроз традиционные методы анализа и мониторинга становятся всё менее эффективными, что создаёт серьёзные уязвимости в системах информационной безопасности. Существующие сигнатурные подходы неспособны справляться с новыми типами атак, особенно тех, которые используют полиморфные вредоносные программы, эксплуатируют уязвимости нулевого дня или применяют скрытные методы проникновения. Это обуславливает необходимость внедрения более интеллектуальных и адаптивных решений, способных эффективно анализировать большие объёмы гетерогенных данных в реальном времени и своевременно выявлять аномалии в сетевом трафике. Представленное исследование направлено на разработку и апробацию современных методов обнаружения аномалий в сетевом трафике на основе алгоритмов машинного обучения, включая как классические, так и продвинутые подходы — глубокие нейронные сети и ансамблевые методы. Цель проекта — создание эффективных моделей, способных автоматически идентифицировать подозрительную активность, классифицировать возможные угрозы и предсказывать потенциальные векторы атак. Особое внимание уделяется практическому применению данных моделей в условиях реального времени без значительного увеличения вычислительных затрат. Для реализации поставленных задач в работе используется аналитическая платформа KNIME (Konstanz Information Miner), которая обеспечивает визуальную среду для построения, обучения и тестирования моделей машинного обучения. Благодаря модульной архитектуре и интуитивно понятному интерфейсу, KNIME упрощает процесс обработки данных, построения рабочих процессов и интерпретации результатов. Это делает платформу особенно привлекательной для специалистов, не обладающих глубокими знаниями в программировании, но заинтересованных в использовании интеллектуального анализа данных для повышения уровня кибербезопасности.

Ключевые слова: машинное обучение, датасет, обработка данных, анализ данных, кибербезопасность, сетевой трафик

Для цитирования: Лапин В. Г., Гальвас И. А., Головин И. В., Кенько Е. А., Васильев М. Н. Разработка методов выявления аномалий сетевого трафика с использованием алгоритмов машинного обучения // Современная наука и инновации. 2025. № 2. С. 51–62. <https://doi.org/10.37493/2307-910X.2025.2.5>

Research article

The development of methods for detecting network traffic anomalies using machine-learning algorithms

Vitaly G. Lapin^{1*}, Ivan A. Galvas², Ivan V. Golovin³, Egor A. Kenkov⁴, Maxim N. Vasiliev⁵

^{1,2,3,4} North-Caucasus Federal University (1, Pushkin St., Stavropol, 355017, Russia)

⁵ MIREA - Russian Technological University (78, Vernadsky Ave., Moscow, 119454, Russia)

¹ vitlx@yandex.ru; <https://orcid.org/0000-0002-0611-7002>

² norra7@yandex.ru

³ ivangolovin085@gmail.com; <https://orcid.org/0009-0008-0512-7242>

⁴ kenkov2045@yandex.ru; <https://orcid.org/0009-0005-3078-1412>

⁵ kesumen_2000@mail.ru

*Corresponding author

Abstract. *With the rapid growth of network traffic and the complexity of the cyber threat structure, traditional methods of analysis and monitoring are becoming less effective, which creates serious vulnerabilities in information security systems. Existing signature-based approaches are unable to cope with new types of attacks, especially those that use polymorphic malware, exploit zero-day vulnerabilities or use stealthy penetration methods. This necessitates the implementation of more intelligent and adaptive solutions that can effectively analyze large volumes of heterogeneous data in real time and promptly detect anomalies in network traffic. The presented study is aimed at developing and testing modern methods for detecting anomalies in network traffic based on machine learning algorithms, including both classical and advanced approaches - deep neural networks and ensemble methods. The goal of the project is to create effective models that can automatically identify suspicious activity, classify possible threats and predict potential attack vectors. Particular attention is paid to the practical application of these models in real time without a significant increase in computational costs. To implement the tasks set, the work uses the KNIME (Konstanz Information Miner) analytical platform, which provides a visual environment for building, training and testing machine learning models. Thanks to its modular architecture and intuitive interface, KNIME simplifies the process of data processing, building workflows and interpreting results. This makes the platform especially attractive to specialists who do not have deep knowledge of programming, but are interested in using intelligent data analysis to improve cybersecurity.*

Keywords: machine learning, dataset, data processing, data analysis, cybersecurity, network traffic

For citation: Lapin VG, Galvas IA, Golovin IV, Kenkov EA, Vasiliev MN. The development of methods for detecting network traffic anomalies using machine-learning algorithms. *Modern Science and Innovations*. 2025;(2):51-62. (In Russ.). <https://doi.org/10.37493/2307-910X.2025.2.5>

Введение. С ростом объемов сетевого трафика традиционные методы обнаружения киберугроз становятся менее эффективными. Это связано с увеличением сложности атак, разнообразием их векторов, а также высокой скоростью распространения вредоносной активности. В таких условиях критически важно внедрять современные подходы к анализу трафика, способные автоматически выявлять аномалии и минимизировать время реагирования на инциденты.

Одним из перспективных направлений в этой области является применение методов машинного обучения (МО), которые позволяют не только обнаруживать отклонения в сетевом трафике, но и классифицировать угрозы, а также прогнозировать потенциальные атаки. Однако разработка и настройка моделей машинного обучения требуют специализированных инструментов, обеспечивающих удобство работы, наглядность и эффективность обработки данных [1–5].

В данном исследовании рассматривается использование платформы KNIME (Konstanz Information Miner) для создания и анализа моделей машинного обучения, направленных на выявление аномалий в сетевом трафике. KNIME предоставляет интуитивно понятный графический интерфейс, что ускоряет разработку рабочих процессов и делает каждый этап анализа более прозрачным.

Актуальность. В условиях цифровой трансформации и роста числа кибератак традиционные сигнатурные методы защиты, основанные на заранее известных шаблонах угроз, демонстрируют свою ограниченность. Современные злоумышленники используют сложные техники обхода защиты, включая полиморфные вредоносные программы, атаки нулевого дня и скрытые методы проникновения в сети. В связи с этим возникает необходимость в более гибких и адаптивных подходах к мониторингу и анализу сетевого трафика.

Машинное обучение предлагает решение этой проблемы за счет способности выявлять скрытые закономерности и аномалии в больших объемах данных. Однако внедрение ML-моделей в практику информационной безопасности требует не только глубоких знаний в области анализа данных, но и удобных инструментов для их реализации. Платформа KNIME, благодаря своей модульности и визуальному подходу к построению рабочих процессов, значительно упрощает процесс разработки, тестирования и развертывания моделей машинного обучения.

Результаты данного исследования могут быть полезны специалистам по кибербезопасности, аналитикам данных и разработчикам систем защиты информации, заинтересованным в автоматизации обнаружения угроз и повышении эффективности мониторинга сетевой активности.

Анализ датасета SNT_dataset.csv. Датасет был сгенерирован с использованием эмуляции сети и инструментов анализа трафика, таких как CICFlowMeter и Wireshark. Данные содержат записи о сетевых потоках, включая:

- Временные характеристики (длительность потока, временные метки).
- Объёмные показатели (количество пакетов, байт).
- Статистические метрики (скорость передачи, распределение пакетов).

Датасет содержит 1034668 записей с 22 признаками, включая данные из таблицы 1:

Таблица 1. Типы сетевых аномалий /
Table 1. Types of Network Anomalies

Признак	Тип данных	Описание
timestamp	Числовой	Временная метка потока
flow_duration_sec	Числовой	Длительность потока в секундах
packet_count	Числовой	Общее количество пакетов
byte_count	Числовой	Общий объём данных в байтах
packet_count_per_second	Числовой	Скорость передачи пакетов
byte_count_per_second	Числовой	Скорость передачи данных
ip_proto	Категориальный	Протокол (TCP/UDP/ICMP)
label	Бинарный	0 – нормальный трафик, 1 – аномалия

Классы несбалансированы: ~85% нормального трафика и ~15% аномалий.

Выбросы: некоторые потоки имеют аномально высокие значения byte_count (возможные DDoS-атаки).

Корреляция признаков: packet_count и byte_count сильно коррелируют (коэффициент Пирсона > 0.9).;

- управление памятью и контекстами по клиенту;
- автоматизация MLOps для каждого арендатора.

На схеме (рис. 1) приведена систематизация основных проблем реализации анализируемого подхода.



Рисунок 1. Схема моделей машинного обучения /
Figure 1. Diagram of machine learning models

Моделирование в KNIME позволяет построить прогнозные модели, которые дают ценную информацию о будущих тенденциях. Платформа подходит для разработки сложных прогнозных моделей.

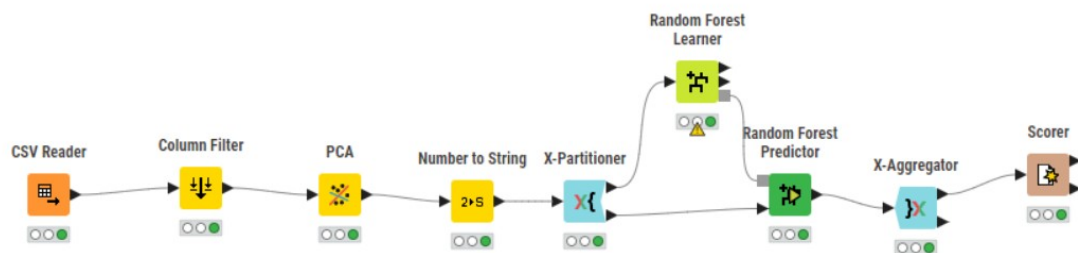


Рисунок 2. Схема моделей машинного обучения /
Figure 2. Machine Learning Models Diagram

В данной схеме используются 8 узлов.

CSV Reader: Читает CSV-файлы. Чтобы автоматически угадать структуру файла, нажмите кнопку автоматического определения формата. Если у вас возникнут проблемы с неправильно угаданными типами данных, отключите опцию. Ограничить количество проверяемых строк данных на вкладке Дополнительные настройки. Если структура

входного файла изменяется между различными вызовами, включите опцию "Поддерживать изменение файловых схем" на вкладке "Дополнительные настройки". Дополнительные сведения см. в Руководстве по обработке файлов KNIME. Примечание: Если вы обнаружите, что этот узел не может прочитать ваш файл, попробуйте использовать узел для чтения файлов. Он предлагает больше возможностей для чтения сложных файлов. Этот узел может обращаться к множеству различных файловых систем.

Column Filter: Этот узел позволяет фильтровать столбцы из входной таблицы, в то время как в выходную таблицу передаются только оставшиеся столбцы. В диалоговом окне столбцы можно перемещать между списками включения и исключения.

PCA: Этот узел выполняет анализ главных компонент (PCA) на заданных данных. Входные данные проецируются из исходного пространства признаков в пространство (возможно) меньшей размерности с минимальной потерей информации.

Number to String: Преобразует числа в столбце (или наборе столбцов) в строки. Обратите внимание, что для расширенной настройки, такой как округление или представление в научном уведомлении, вы также можете использовать узел "Округление вдвое".

X-Partitioner: Этот узел является первым в цикле перекрестной проверки. В конце цикла должен быть X-агрегатор для сбора результатов каждой итерации. Все узлы, находящиеся между этими двумя узлами, выполняются столько раз, сколько итераций должно быть выполнено. **ML_Model Learner:** здесь **ML_Model** обозначает любую модель машинного обучения, например, дерево решений, случайный лес, логистическую регрессию и т. д. **Learner** означает, что модель обучается на обучающих данных.

X-Aggregator: Этот узел должен быть концом цикла перекрестной проверки и должен следовать за узлом X-Partitioner. Он собирает результат из узла-предиктора, сравнивает предсказанный класс и реальный класс и выводит прогнозы для всех строк и статистику итераций.

Scorer: Сравнивает два столбца по парам значений атрибутов и показывает матрицу путаницы, т.е. сколько строк какого атрибута и их классификация совпадают. Кроме того, можно выделить ячейки этой матрицы, чтобы определить базовые строки. Диалоговое окно позволяет выбрать два столбца для сравнения; значения из первого выбранного столбца отображаются в строках матрицы путаницы, а значения из второго столбца - в столбцах матрицы путаницы. Результатом работы узла является матрица путаницы с количеством совпадений в каждой ячейке. Кроме того, второй внешний порт предоставляет ряд статистических данных о точности, таких как истинно положительные, ложноположительные, Истинно отрицательные, Ложноотрицательные результаты, Отзыв, точность, чувствительность, специфичность, F-мера, а также общая точность и коэффициент каппы Козна.

Random Forest Learner состоит из выбранного количества деревьев решений. Каждое из деревьев решений строится на основе отдельного набора строк (записей), и для каждого разделения внутри дерева используется случайно выбранный набор столбцов (описывающих атрибутов). Наборы строк для каждого дерева решений создаются методом начальной загрузки и имеют тот же размер, что и исходная входная таблица. Набор атрибутов для отдельного разделения в дереве решений определяется случайным выбором атрибутов $\sqrt{m}$ из доступных атрибутов, где m - общее количество обучающих столбцов. Атрибуты также могут быть предоставлены в виде бита (отпечатка пальца), байта или двойного вектора. Выходная модель описывает случайный лес и применяется в соответствующем узле прогнозирования.

Этот узел предоставляет часть функций обучающего ансамбля деревьев, соответствующую случайному лесу.

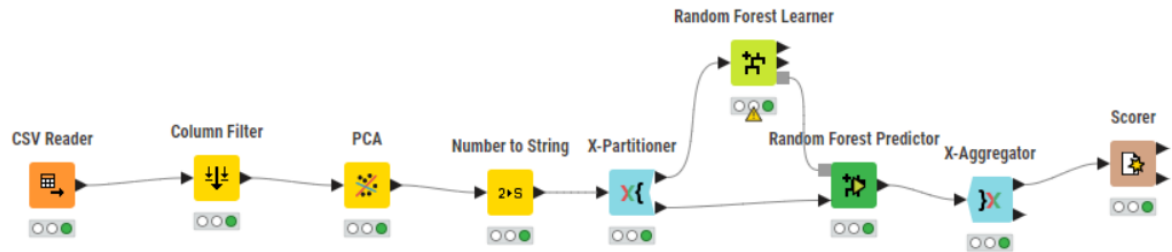


Рисунок 3. Схема модели машинного обучения с использованием Random Forest Learner /
Figure 3. Diagram of a machine learning model using Random Forest Learner

Исследование с использованием блока X-Partitioner.

Таблица 2. Анализ модели с использованием блока X-Partitioner /
Table 2. Model analysis using the X-Partitioner block

Number of validation	Точность
2	70,295%
5	70,295%
7	81,771%
8	81,79%

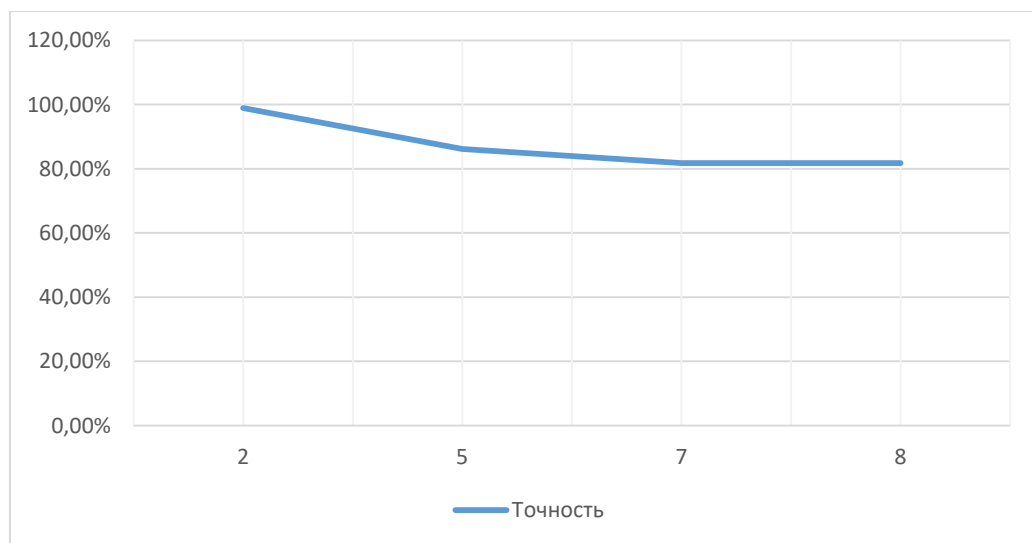


Рисунок 4. Изменение точности модели /
Figure 4. Changing the accuracy of the model

Linear Regression Learner.

Этот узел выполняет многомерную линейную регрессию. Выберите в диалоговом окне целевой столбец (поле со списком сверху), т. е. ответ. Два списка в центре диалогового окна позволяют включать только определенные столбцы, представляющие (независимые) переменные. Убедитесь, что столбцы, которые вы хотите включить, находятся в правильном списке «включено». В статистике линейная регрессия — это модель, которая оценивает связь между скалярным откликом (зависимой переменной) и одной или несколькими объясняющими переменными (регрессором или независимой переменной).

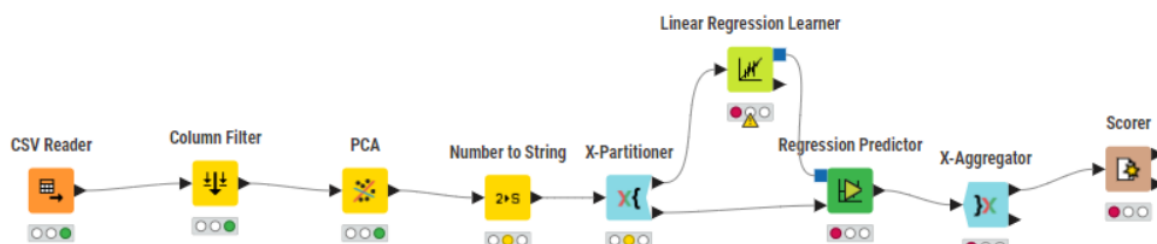


Рисунок 5. Схема модели машинного обучения с использованием Regression Learner /
Figure 5. Diagram of a machine learning model using Regression Learner

Исследование с использованием блока X-Partitioner.

Таблица 3. Анализ модели с использованием блока X-Partitioner /
Table 3. Model analysis using the X-Partitioner block

Number of validation	Точность
2	75,001%
4	62,499%
6	58,332%
8	56,25%

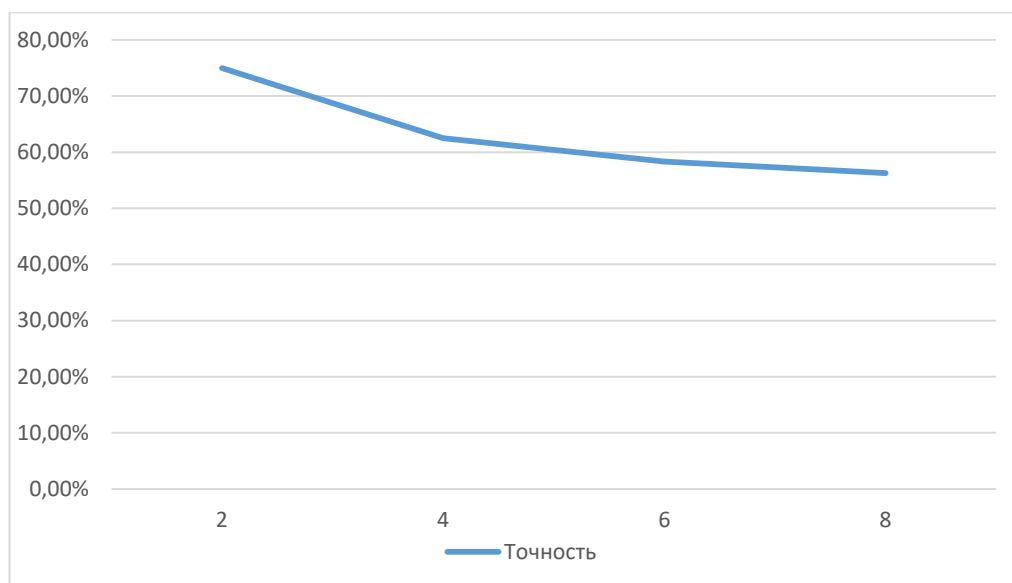


Рисунок 6. Изменение точности модели /
Figure 6. Changing the accuracy of the model

Decision Tree Learner.

Этот узел индуцирует дерево решений классификации в основной памяти. Целевой атрибут должен быть номинальным. Другие атрибуты, используемые для принятия решений, могут быть как номинальными, так и числовыми. Числовые разделения всегда являются бинарными (два результата), разделяя домен на два раздела в заданной точке разделения. Номинальные разделения могут быть либо бинарными (два результата), либо иметь столько же результатов, сколько и номинальных значений. В случае бинарного разделения номинальные значения делятся на два подмножества.

Алгоритм предоставляет две меры качества для расчета разделения: индекс Джини и коэффициент усиления. Кроме того, существует метод пост-обрезки для уменьшения размера дерева и повышения точности прогнозирования. Метод обрезки основан на принципе минимальной длины описания.

Алгоритм может быть запущен в несколько потоков и, таким образом, использовать несколько процессоров или ядер.

Большинство методов, используемых в этой реализации дерева решений, можно найти в "Программах C4.5 для машинного обучения" Дж. Р. Куинлана и в "SPRINT: масштабируемый параллельный классификатор для интеллектуального анализа данных" Дж. Шафера, Р. Агравала, М. Мехты.

Naïve Bayes Learner.

Этот Узел создает байесовскую модель из заданных обучающих данных. Он вычисляет количество строк на значение атрибута на класс для номинальных атрибутов и гауссово распределение для числовых атрибутов. Созданная модель может использоваться в наивном байесовском предсказателе для прогнозирования принадлежности к классу неклассифицированных данных.

Узел отображает предупреждающее сообщение, если какие-либо столбцы игнорируются из-за неподдерживаемых типов данных. Например, столбцы Bit Vector игнорируются, когда включен флаг совместимости PMML, поскольку они не поддерживаются стандартом PMML. Алгоритм может быть запущен в несколько потоков и, таким образом, использовать несколько процессоров или ядер.

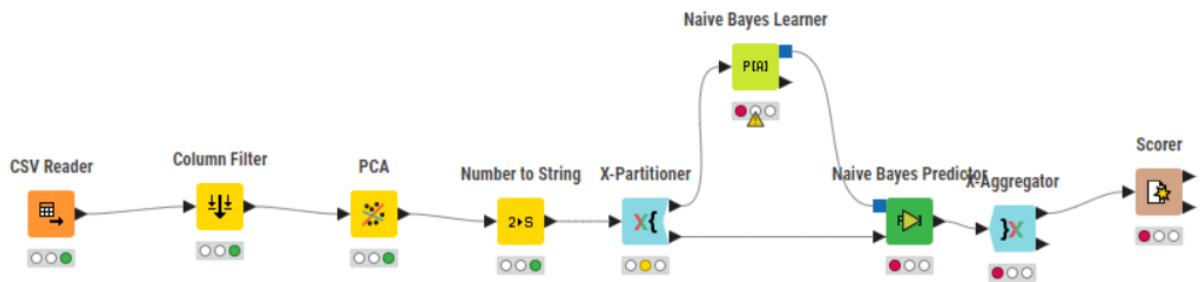


Рисунок 7. Схема модели машинного обучения с использованием Naïve Bayes Learner /
Figure 7. Diagram of a machine learning model using Naïve Bayes Learner

Исследование с использованием X-Partitioner.

Таблица 4. Анализ модели с использованием блока X-Partitioner /
Table 4. Model analysis using the X-Partitioner block

Limit number of levels (tree depth)	Точность
15	68,807%
17	68,788%
19	68,829%
21	68,967%

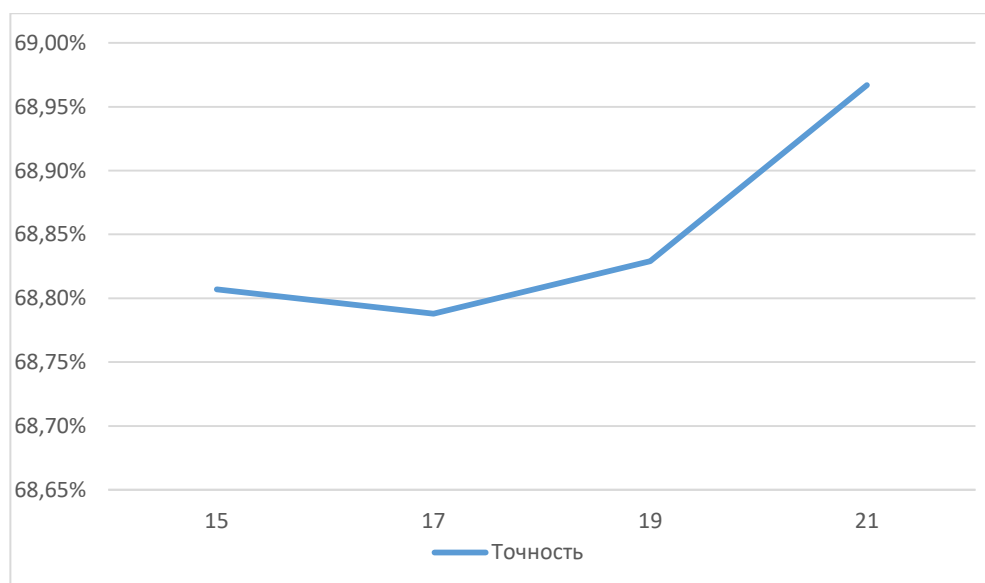


Рисунок 8. Изменение точности модели /
Figure 8. Changing the accuracy of the model

Gradient Boosted Trees Learner изучает деревья с градиентным усилением с целью классификации. Алгоритм использует очень мелкие деревья регрессии и специальную форму усиления для построения ансамбля деревьев. Реализация следует алгоритму из раздела 4.6 статьи "Greedy Function Approximation: A Gradient Boosting Machine" Джерома Х. Фридмана (1999).

Используемый базовый обучающийся для этого ансамблевого метода – это простое дерево регрессии, как оно используется в узлах Tree Ensemble, Random Forest и Simple Regression Tree.

По умолчанию дерево строится с использованием бинарных расщеплений для числовых и номинальных атрибутов (последнее можно изменить на многопутевые расщепления). Встроенная обработка пропущенных значений пытается найти лучшее направление для пропущенных значений, проверяя каждое возможное направление и выбирая то, которое дает наилучший результат (т. е. наибольший прирост).

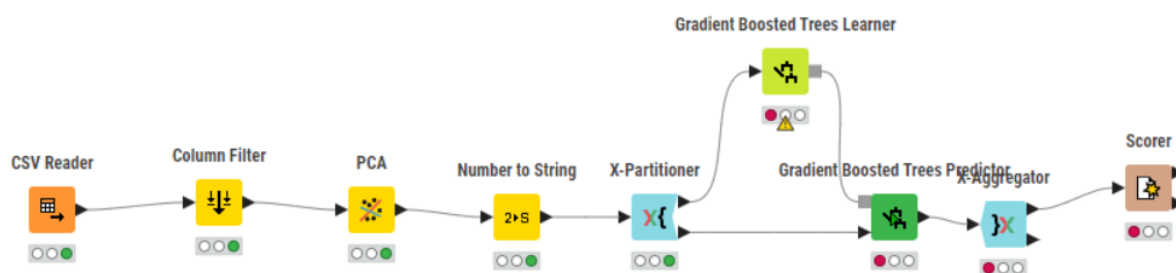


Рисунок 9. Схема модели машинного обучения с использованием Gradient Boosted Trees Learner /
Figure 9. Diagram of a machine learning model using Gradient Boosted Trees Learner

Исследование с использованием X-Partitioner

Таблица 5. Анализ модели с использованием блока Limit number of levels (tree depth) /
Table 5. Model analysis using the Limit number of levels (tree depth) block

Limit number of levels (tree depth)	Точность
15	54,807%
17	65,788%
19	78,829%
21	89,967%

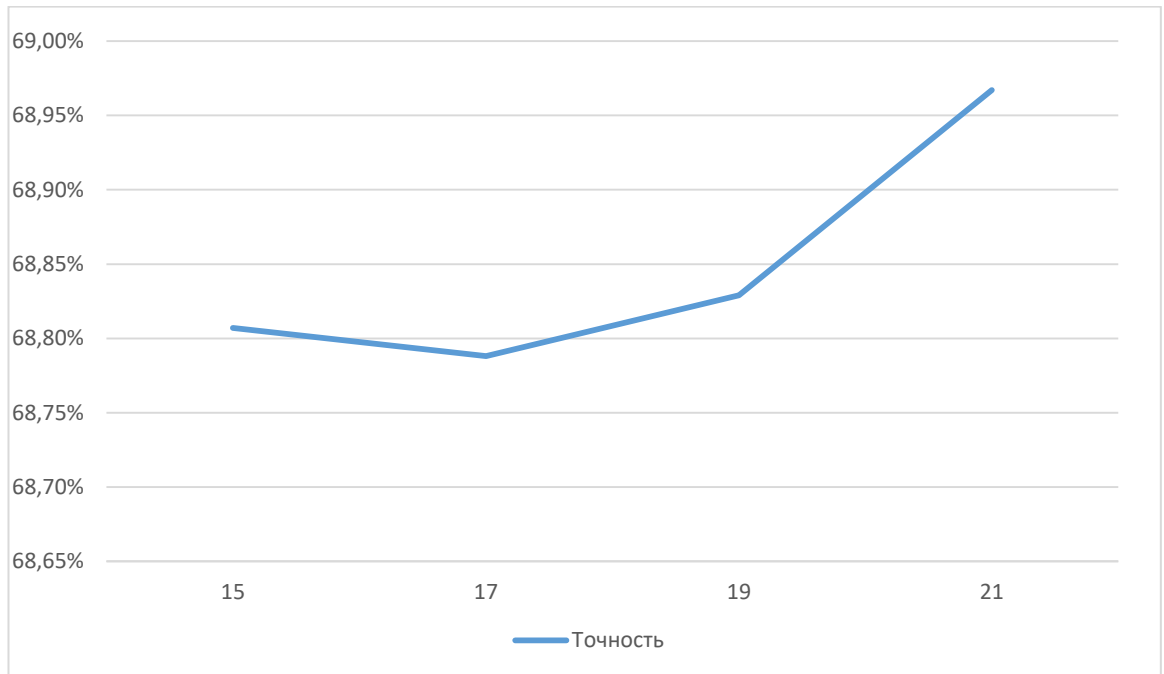


Рисунок 10. Изменение точности модели /
Figure 10. Changing the accuracy of the model

В дереве регрессии прогнозируемое значение для листового узла – это среднее целевое значение записей в листе. Следовательно, прогнозы являются наилучшими (по отношению к обучающим данным), если дисперсия целевых значений в листе минимальна. Это достигается путем разбиений, которые минимизируют сумму квадратов ошибок в соответствующих потомках.

Большинство методов, используемых в этой реализации дерева решений, можно найти в "Программах C4.5 для машинного обучения" Дж. Р. Куинлана и в "SPRINT: масштабируемый параллельный классификатор для интеллектуального анализа данных" Дж. Шафера, Р. Агравала, М. Мехты.

Заключение. Разработанные в рамках данного проекта методы обнаружения аномалий сетевого трафика с использованием алгоритмов машинного обучения демонстрируют значительный потенциал для повышения эффективности современных систем кибербезопасности. Проведенное исследование подтвердило, что традиционные подходы к анализу сетевой активности уступают ML-решениям в точности, скорости обработки данных и способности выявлять сложные, ранее неизвестные угрозы.

ЛИТЕРАТУРА

1. Греков М. М. Учебный стенд для анализа методов обнаружения аномалий на основе теории машинного обучения // Моделирование, оптимизация и информационные технологии. 2022. Т. 10. № 1. С. 22. <https://doi.org/10.26102/2310-6018/2022.36.1.019>

2. Поздняк И. С., Макаров И. С. Модели обнаружения атак с использованием методов машинного обучения // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2024. № 1. С. 99–109. <https://doi.org/10.18137/RNU.V9187.24.01.P.99>
3. Каскырбаева К. С. Применение методов машинного обучения для выявления аномалий в сетевом трафике // Вестник науки. 2025. Т. 2. № 4. С. 712–717.
4. Татарникова Т.М., Богданов П.Ю. Обнаружение атак в сетях интернета вещей методами машинного обучения // Информационно-управляющие системы. 2021. № 6 (115). С. 42–52. <https://doi.org/10.31799/1684-8853-2021-6-42-52>
5. Мурат С.А. Обнаружение и предотвращение кибератак с помощью гибридного подхода: использование машинного обучения и традиционных методов // Студенческий вестник. 2024. № 37-4 (323). С. 63–67.

REFERENCES

1. Grekov MM. A training device for the analysis of anomaly detection methods based on machine learning theory. Modeling, optimization and information technology. 2022; 10(1):22. (In Russ.). <https://doi.org/10.26102/2310-6018/2022.36.1.019>
2. Pozdnyak IS, Makarov IS. Attack detection models using machine learning methods. Vestnik of Russian new University. Series: Complex systems: Models, analysis, management. 2024;1:99-109. (In Russ.). <https://doi.org/10.18137/RNU.V9187.24.01.P.99>
3. Kaskyrbaeva KS. The use of machine learning methods to identify anomalies in network traffic. Science bulletin. 2025; 2(4):712-717. (In Russ.).
4. Tatarnikova TM, Bogdanov PY. Intrusion detection in internet of things networks based on machine learning methods. Information and control systems. 2021;6(115):42-52. (In Russ.). <https://doi.org/10.31799/1684-8853-2021-6-42-52>
5. Murat SA. Detection and prevention of cyber-attacks using a hybrid approach: using machine learning and traditional methods. Student Bulletin. 2024;37-4(323):63-67. (In Russ.).

ИНФОРМАЦИЯ ОБ АВТОРАХ

Виталий Геннадьевич Лапин – доцент кафедры вычислительной математики и кибернетики, Северо-Кавказский Федеральный Университет, vitlx@yandex.ru

Иван Алексеевич Гальвас - студент кафедры вычислительной математики и кибернетики, Северо-Кавказский Федеральный Университет, norra7@yandex.ru

Иван Владимирович Головин - студент кафедры вычислительной математики и кибернетики, Северо-Кавказский Федеральный Университет, ivangolovin085@gmail.com

Егор Александрович Кеньков - студент кафедры вычислительной математики и кибернетики, Северо-Кавказский Федеральный Университет, kenkov2045@yandex.ru

Максим Николаевич Васильев - студент кафедры вычислительной математики и кибернетики, Северо-Кавказский Федеральный Университет, kesumen_2000@mail.ru

Вклад авторов: все авторы внесли равный вклад в подготовку публикации.

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Статья поступила в редакцию: 01.04.2025;
одобрена после рецензирования: 13.05.2025;
принята к публикации: 01.06.2025.

INFORMATION ABOUT THE AUTHORS

Vitaliy G Lapin – Associate Professor, Department of Computational Mathematics and Cybernetics, North Caucasus Federal University, vitlx@yandex.ru

Ivan A. Galvas – Student, Department of Computational Mathematics and Cybernetics, North Caucasus Federal University, norra7@yandex.ru

Ivan V. Golovin – Student, Department of Computational Mathematics and Cybernetics, North Caucasus Federal University, ivangolovin085@gmail.com

Egor A. Kenkov – Student, Department of Computational Mathematics and Cybernetics, North Caucasus Federal University, kenkov2045@yandex.ru

Maxim N. Vasiliev – Student, Department of Computational Mathematics and Cybernetics, North Caucasus Federal University, kesumen_2000@mail.ru

Contribution of the authors: the authors contributed equally to this article.

Conflict of interest: the authors declare no conflicts of interests.

The article was submitted: 01.04.2025;

approved after reviewing: 13.05.2025;

accepted for publication: 01.06.2025.